

Introduction aux objets

F.-Y. Villemin, CNAM



[http://deptinfo.cnam.fr/Enseignement/
CycleSpecialisation/MAI/index.html](http://deptinfo.cnam.fr/Enseignement/CycleSpecialisation/MAI/index.html)

Plan

- Définitions
- Evolution de la programmation
- Méthodes Classiques
- Objet
- Concepts objets
- Méthodologies orientée-objet
- Problèmes

Définitions

Méthodologie :

Entrée: une spécification floue, réduite,
éventuellement inconsistante du projet

Sortie: une description complète, consistante,
compréhensible

- des caractéristiques
- du comportement
- du (des buts) du projet

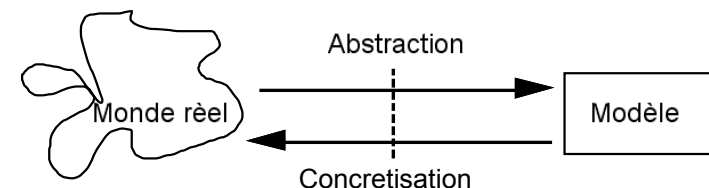
Définitions

Abstraction (Larousse)

Résultat d'abstraire; isoler une propriété, un caractère d'un objet en le considérant à part.

Modèle (Larousse)

Toute structure formalisée utilisée pour rendre compte d'un ensemble de phénomènes qui possèdent entre eux certaines relations.



Définitions

Système (Larousse)

- (1) Combinaison d'éléments qui se coordonnent pour concourir à un résultat ou de manière à former un ensemble.
- (2) Ensemble de méthodes, de procédés destinés à produire un résultat.

Concept

Idée d'un objet conçu par l'esprit permettant d'organiser les perceptions et les connaissances.

Concept

Pour les nominalistes, seuls les individus et les populations d'individus existent; les catégories sont des abstractions construites par l'homme correspondant à des associations, des agrégations d'individus...

Ockham : "il n'y a que des individus", toute classification est artificielle et est due à l'homme

Buffon : "les genres, les classes, les ordres n'existent que dans notre imagination"

Concept

Les choses sont toutes individuelles et singulières

Un concept est une abstraction permettant de classer les choses qui résulte d'une généralisation de cas particuliers :

Ex: la couleur, la forme

Les concepts sont les produits d'actes mentaux, comme penser, juger, raisonner...

Les concepts ne sont que des "dénominations", des mots, des signes, des termes conventionnels, des représentations n'ayant pas de réalité substantielle

Un concept est un mot qui comme tout mot qui comme tout mot peut être combiné avec d'autres mots pour créer des phrases pourvues de sens (P. King)

Modèle

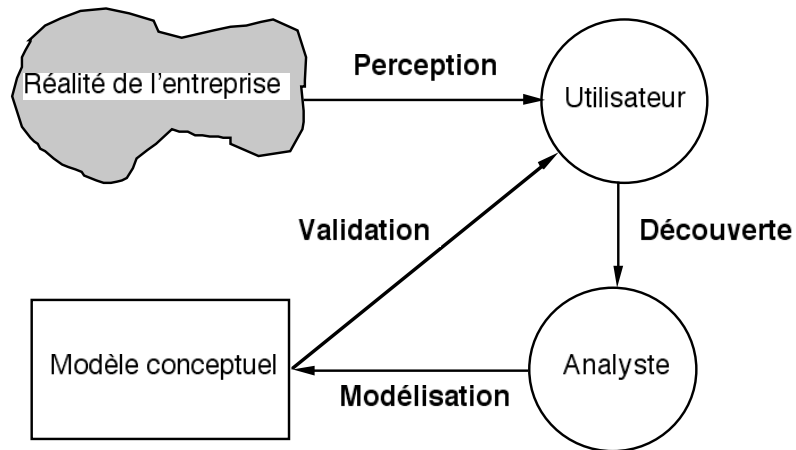
Un **modèle** associe un énoncé formel dans un langage formel à une "réalité empirique"

L'interprétation d'un modèle est l'association d'un sens aux symboles du langage formel de telle sorte que cet énoncé soit vrai

Minsky : Un objet *O* est un modèle d'une réalité *R* si *O* permet de répondre aux questions que l'on se pose sur *R*

Modèle

Processus de détermination du modèle conceptuel



Objet

Objet (Larousse)

(1) Chose définie par sa matière, sa forme, sa couleur.

(2) But d'une action.

philosophie: ce qui est perçu ou pensé et s'oppose à l'être pensant ou le sujet

Guillaume d'Occam (1320) considère que **seuls les objets sont des réalités**, et que les espèces, **les catégories** etc. ne sont pas des réalités mais **des constructions de l'esprit**, des abstractions, de simples mots (d'où l'appellation de nominalisme)

Il n'y a donc **pas de catégorie observable**

Evolution de la programmation

Wirth: Algorithms + Data Structures = Programmes

Programme = {données} + {procédures agissant sur les données}

→ Méthodologies :

- **programmation structurée**
 - Dirigées par les traitements
 - Diviser pour régner
 - Approche modulaire (procédures, modules)
- **contrôle statique des types**

Inconvénients:

Programmes figés (évolution de l'environnement)

Perte de la sémantique du système (buts, comportements...)

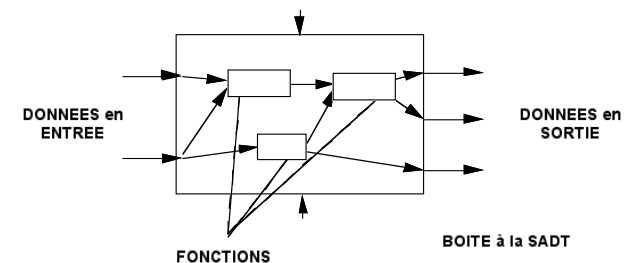
Méthodes Classiques

Spécification du système complet

Caractérisation de la **totalité des données et traitements**

Plusieurs vues:

- Entité/association → MERISE
- Fonctions → SADT ou SA/RT



Objet

Un **objet** est un(e):

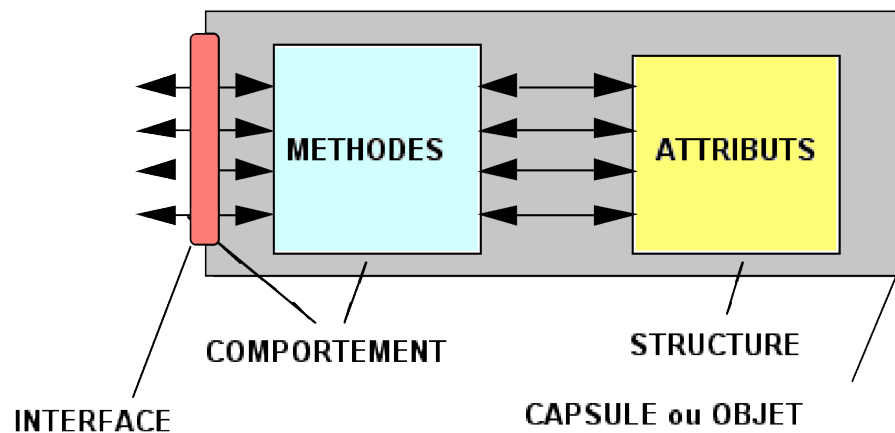
- **unité de base de modularisation** (en structure et en comportement)
- concept qui se réfère à une chose (réelle ou non) identifiable par l'utilisateur
- a des caractéristiques décrites par son état (local)
- a des opérations qui changent le système localement (réactions en chaîne)

Objet

Un **objet** a :

- une **identité** (OID)
- une **liste d'états** (qu'il rencontre au cours de sa vie)
- un **comportement**
- des opérations qu'il peut faire sur d'autres objets
- des opérations que les autres objets peuvent faire sur lui.

Objet



Concepts objets

Encapsulation

- regroupement des données et des procédures agissant sur ces données → objet
- objet connu par son interface → comportement
- notion de données et procédures privées

Abstraction de données

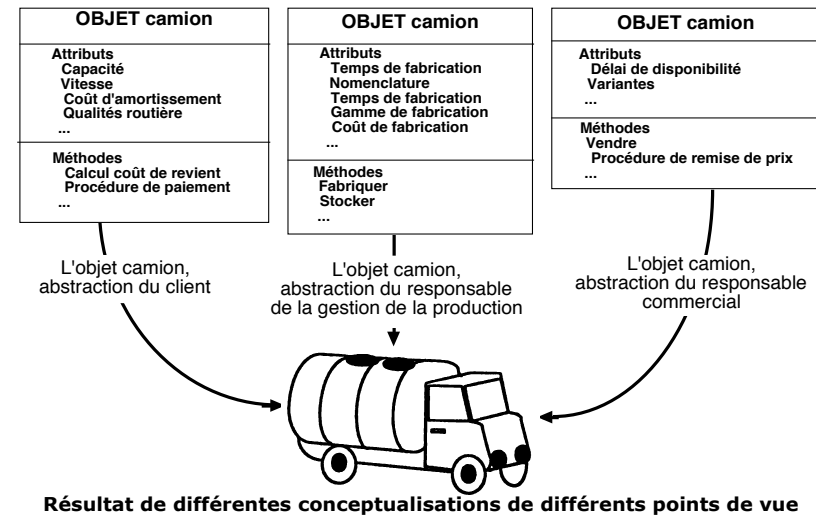
- procédure = méthode
- appel de procédure = accès à une BD
- message → requête, mise à jour
message = <objet destinataire, méthode, paramètres>
- implantation cachée

Concepts objets

Généralisation

- regroupement d'objets ayant un comportement similaire → **classes** (généralisation)
- **classes** → propriétés et caractéristiques des objets similaires
- classe de classes
 - sous-classes, super-classes
 - hiérarchie (réseau) de classes
 - relation "classe-de"

Concepts objets



Concepts objets

Spécialisation

- objet d'une classe = relation "est-une-instance-de"
- instanciation : classe → objet

Agrégation

- réunion d'objets formant un objet
 - relation "est-une-partie-de"
 - relation "est-composé-de"

Concepts objets

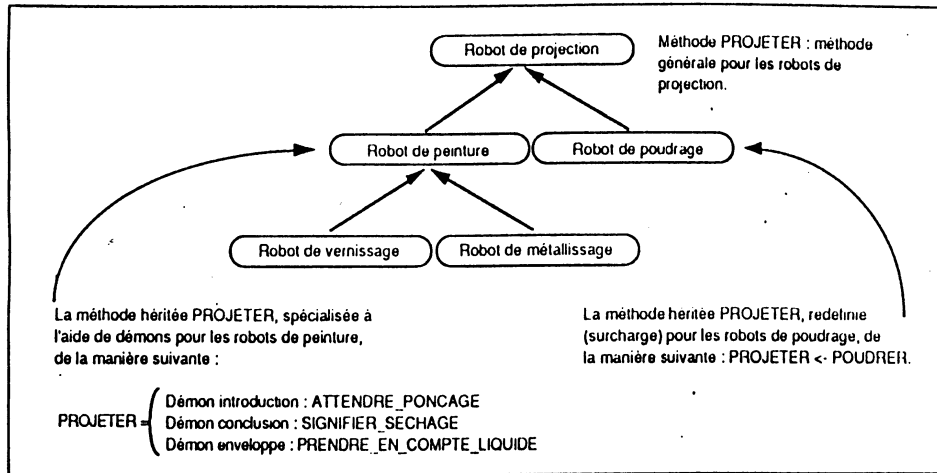
Héritage

- méthode qui permet de décrire une classe comme une spécialisation d'une classe existante
 - un objet d'une sous-classe **hérite de toutes les propriétés et attributs** de la classe mère
- relation "hérite-de"
- héritage simple ou multiple

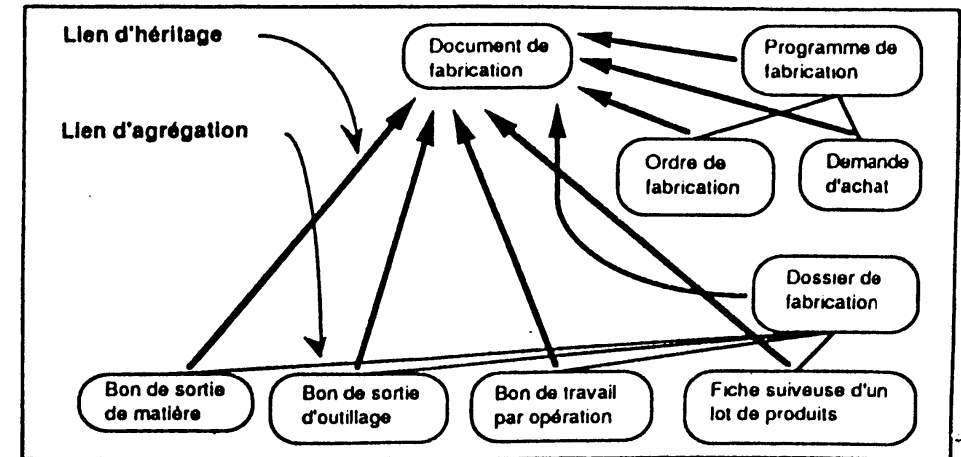
Surcharge d'une méthode

- même méthode mais comportements différents suivant objets

Concepts objets



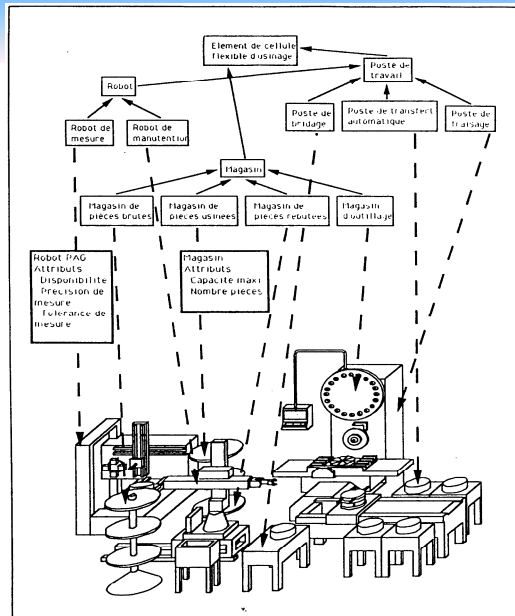
Concepts objets



Différence entre liens d'héritage et liens d'agrégation pour les documents nécessaires à une gestion de production utilisant le concept M.R.P. (Manufacturing Resource Planning).

Concepts objets

Un modèle
d'atelier
flexible



Méthodologies orientées-objet

- Spécification comme un **système ouvert**
- **Interface avec l'existant** (communication avec l'homme)
- Caractérisation du **comportement ou du but** du système
- Décomposition en **sous-systèmes ou objets**
- Plusieurs vues :
 - Comportement → MOO
 - Buts → KADS

Méthodologies orientées-objet

Principe de continuité :

- les mêmes concepts sont utilisés tous au long du cycle de vie : Analyse, Conception, et Programmation "objet"
- L'objet est un sous système (Données + Procédures)
- Pas de décomposition systémique possible
- Modélisation des objets du système d'information, mais pas du système d'information lui-même

Méthodologies orientées-objet

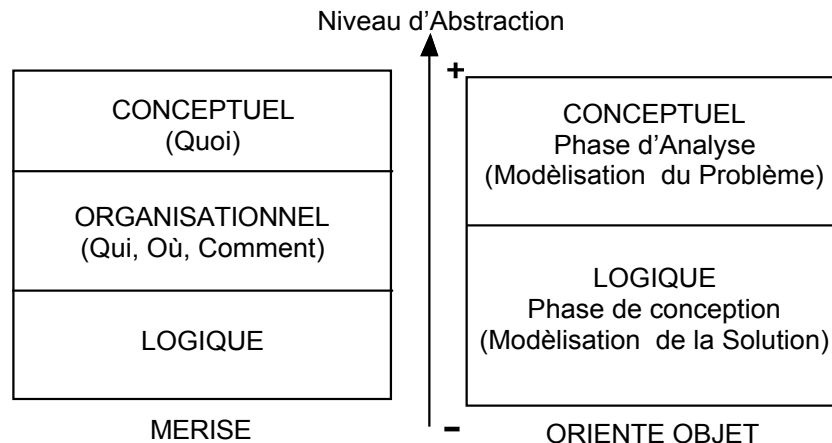
Principe de la réutilisabilité

Autorise une meilleure gestion de la complexité :

- réduction du nombre de noms par l'encapsulation
- classification des éléments du logiciel (utilisation de classes)
- suppression des redondances (héritage ou délégation)

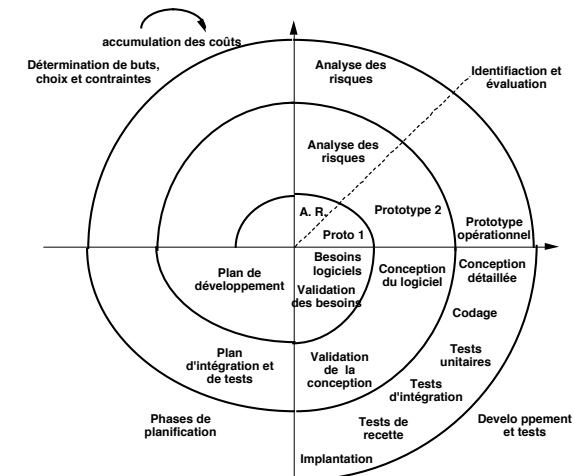
Méthodologies orientées-objet

Comparaison des méthodes



Méthodologies orientées-objet

Cycle de vie en spirale (Boehm 88)



Méthodologies orientées-objet

Modifiabilité

du fait de l'encapsulation et de l'abstraction des données

Réutilisabilité

bibliothèque de classes (d'objets-métier)

construction d'objets

- par instanciation
- par composition d'objets

Lisibilité

détails d'implantation cachés

interfaces des objets = mode d'emploi des objets

→ IDL de CORBA

Problèmes

(1) Description du comportement

plus simple au plus compliqué

objet complexe à partir d'objets simples

caractérisation du comportement de l'objet complexe?

plus compliqué au plus simple

pour grande équipe

résultats difficilement réutilisables

(2) Déclaratif (quoi) ou procédural (comment)?

UML

UML ou Unified Modeling Language

Langage graphique de description de systèmes objets

Version XML : XMI permet l'échange entre ateliers de divers éditeurs



UML - Motivations

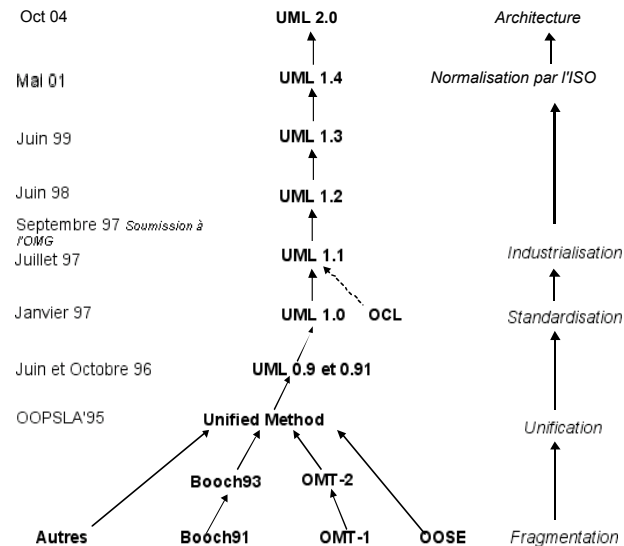
UML n'est une méthodologie orientée objet, mais un langage de description des objets

UML a été créé par James Rumbaugh, Grady Booch et Yvar Jacobson

UML est un langage de modélisation rigoureux pour la modélisation de systèmes complexes :

- Concepts précis
- Sémantique claire
- Notations non ambiguës
- Langage de description de contraintes : OCL

UML - Historique



UML - Modèles

UML définit 9 types de diagrammes :

- Les diagrammes de classes
- Les diagrammes de séquence
- Les diagrammes de collaboration
- Les diagrammes d'objet
- Les diagrammes d'états-transitions
- Les diagrammes d'activités
- Les diagrammes des cas d'utilisation
- Les diagrammes de composants
- Les diagrammes de déploiement

UML - Modèles

UML définit plusieurs modèles pour la représentation des systèmes :

- Le **modèle de classes** capture la **structure statique**
- Le modèle des états exprime le **comportement de classes**
- Le **modèle des cas d'utilisation** décrit les **besoins des utilisateurs**
- Le **modèle d'interaction** représente les **scénarii** et les **flots de messages**
- Le **modèle de réalisation** décrit les **unités de travail**
- Le **modèle de déploiement** précise la **répartition des processus**

Références

- Pierre-Alain Muller & Nathalie Gaertner: *"Modélisation objet avec UML"*, Eyrolles 2000
- "A Rational Approach to Software Development Using Rose 4.0", Documentation Rational Rose 4.0
- "Unified Modeling Language" version 1.4, mai 2001 : <http://www.omg.org>

Ateliers libres (en java)

- argoUML : <http://argouml.tigris.org>
- PoseidonCE : <http://www.gentleware.com>